Replication: An Alternative Softmax Operator for Reinforcement Learning

Chan, Kai-Chieh Huang Po-Wei Peng, Pei-Chiun Liao, Wei-Chen
Department of Computer Science
National Yang Ming Chiao Tung University
{kai9988ckc.cs07,a0716084.cs07,lollypeng100.ee07,wcl.cs07}@nycu.edu.tw

1 Problem Overview

This paper (Asadi and Littman [2017]) introduces a new softmax operator during action selection, which can be used in a SARSA algorithm that computes a Boltzmann policy with a state-dependent temperature parameter. The algorithm is convergent and serves as an alternative to the original Boltzmann policy. The main research problem that is tackled by the paper is that all the common using softmax operator are not idea operator. Some are not non-expansion, some are not differentiable. Hence, The author proposed Mellowmax softmax operator that is proved to be an idea operator and generate a well trade-off between exploration and exploitation during action selection.

2 Background and The Algorithm

An ideal softmax operator is a parameterized set of operators that:

- has parameter settings that allow it to approximate maximization arbitrarily accurately to perform reward-seeking behavior;
- is a non-expansion for all parameter settings ensuring convergence to a unique fixed point;
- is differentiable to make it possible to improve via gradient-based optimization; and
- avoids the starvation of non-maximizing actions.
- Let $X = x_1, ..., x_n$ be a vector of values. We define the following operators:

$$\max(\mathbf{X}) = \max_{i \in \{1,\ldots,n\}} x_i \ ,$$

$$\text{mean}(\mathbf{X}) = \frac{1}{n} \sum_{i=1}^{n} x_i \ ,$$

$$\text{eps}_\epsilon(\mathbf{X}) = \epsilon \ \text{mean}(\mathbf{X}) + (1-\epsilon) \max(\mathbf{X}) \ ,$$

$$\text{boltz}_\beta(\mathbf{X}) = \frac{\sum_{i=1}^{n} x_i \ e^{\beta x_i}}{\sum_{i=1}^{n} e^{\beta x_i}} \ .$$

The Boltzmann operator $boltz_\beta(X)$ is differentiable. It also approximates max as $\beta \rightarrow \infty$and mean as $\beta \rightarrow 0$. However, it is not a non-expansion operator, and therefore, the lack of the non-expansion property leads to multiple fixed points and ultimately a misbehavior in learning and planning.

Mellowmax

Author presents a new softmax operator that is similar to the Boltzmann operator yet is a non-expansion operator. Author also proves several critical properties of this new operator, introduce a new softmax policy, and present empirical results. The alternative Mellowmax softmax operator is defined as follows:

$$\text{mm}_\omega(\mathbf{X}) = \frac{\log(\frac{1}{n}\sum_{i=1}^{n} e^{\omega x_i})}{\omega} \, ,$$

Author proves that $mm\omega$ is a non-expansion operator, and therefore, GVI and SARSA under $mm\omega$ are guaranteed to converge to a unique fixed point. Furthermore, the operator acts more and more like pure maximization as the value of ω is increased. Conversely, as ω goes to $-\infty$, the operator approaches the minimum. And as ω gets closer to zero, $mm_\omega(x)$ approaches the mean of the values in X.

Mellowmax Policy

Author formally defines the maximum entropy Mellowmax policy of a state s as:

$$\pi_{\text{mm}}(s) = \underset{\pi}{\text{argmin}} \sum_{a \in \mathcal{A}} \pi(a|s) \log\big(\pi(a|s)\big) \qquad (2)$$

$$\text{subject to} \begin{cases} \sum_{a \in \mathcal{A}} \pi(a|s)\hat{Q}(s,a) = \text{mm}_\omega(\hat{Q}(s,.)) \\ \pi(a|s) \geq 0 \\ \sum_{a \in \mathcal{A}} \pi(a|s) = 1 \, . \end{cases}$$

After using the method of Lagrange multipliers to solve this system of equations, the probability of taking an action under the maximum entropy Mellowmax policy has the form:

$$\pi_{mm}(a|s) = \frac{e^{\beta \hat{Q}(s,a)}}{\sum_{a \in \mathcal{A}} e^{\beta \hat{Q}(s,a)}} \quad \forall a \in \mathcal{A} \, ,$$

where β is a value for which:

$$\sum_{a \in \mathcal{A}} e^{\beta\left(\hat{Q}(s,a) - \text{mm}_\omega \hat{Q}(s,.)\right)} \left(\hat{Q}(s,a) - \text{mm}_\omega \hat{Q}(s,.)\right) = 0 \, .$$

The argument for the existence of a unique root is simple. As $\beta \to \infty$, the term corresponding to the best action dominates, and so, the function is positive. Conversely, as $\beta \to -\infty$, the term corresponding to the action with lowest utility dominates, and so the function is negative. Finally, by taking the derivative, it is clear that the function is monotonically increasing, allowing us to conclude that there exists only a single root. Therefore, we can find β easily using any root-finding algorithm.

This policy has the same form as Boltzmann softmax, but with a parameter β whose value depends indirectly on ω. This mathematical form arose not from the structure of $mm\omega$, but from maximizing the entropy. One way to view the use of the Mellowmax operator, then, is as a form of Boltzmann policy with a temperature parameter chosen adaptively in each state to ensure that the non-expansion property holds.

3 Detailed Implementation

3.1 Simple MDP

3.1.1 Simple MDP - SARSA

The pseudocode shows how we run SARSA in the simple MDP environtment.

Algorithm 1 Simple MDP - SARSA

```
If using Mellowmax, replace ‘Boltzmann’ with ‘mm’ and relpace $\beta$ with $\omega$.
**Input:** init $Q(s, a) \forall s \in \mathcal{S} \forall a \in \mathcal{A}$, $\alpha$ and $\beta$
for each episode do
    Init $s$
    $a \sim$ Boltzmann with $\beta$
    repeat
        Take action $a$, observe $r, s'$
        $a' \sim$ Boltzmann with $\beta$
        $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$
        $s \leftarrow s'$,$a \leftarrow a'$
    until $s$ is terminal
end for
```

3.1.2 Simple MDP - GVI

The pseudocode below shows how we run generalized value iteration (GVI) in the simple MDP environtment to find the number of fixed points. We don’t know how the paper’s authors determine whether > 1 fixed points. The way we adopt to decide if > 1 fixed points is to try different initial Q values. If some initial Q values lead to different fixed points, it means > 1 fixed points. Hence, it’s more accurate if more various Q initial values are tried.

Algorithm 2 Simple MDP - GVI

```
Consider $s_1$’s two actions’ Q values: $Q(s_1, \mathrm{a})$ and $Q(s_1, \mathrm{b})$
Note that a means action a of $s_1$; $a$ indicates ‘action‘.
**Input:** softmax operator $\otimes$ (bolz or mm), all pairs of $Q(s_1, \mathrm{a}), \hat{Q}(s_1, \mathrm{b})$ to run, $\alpha$ and $\beta$ or $\omega$
$\mathcal{Q} \leftarrow \{\}$                         ▷ $\mathcal{Q}$ stores each trial’s convergence fixed point.
$\delta \leftarrow 10^{-14}$
for each initial value pair $Q(s_1, \mathrm{a}), Q(s_1, \mathrm{b})$ do
    repeat
        for $a \in \{\mathrm{a}, \mathrm{b}\}$ do
            $Q_{\text{copy}} \leftarrow Q(s_1, a)$
            $Q(s_1, a) \leftarrow R(s, a) + \gamma \sum_{s' \in \{s_1, s_2\}} P(s, a, s') \otimes Q(s', \cdot)$
            $\text{diff}_a \leftarrow \max(\text{diff}_a, |Q_{\text{copy}} - Q(s_1, a)|)$
        end for
    until $\max(\text{diff}_a) < \delta$
    Push fixed point $(Q(s_1, \mathrm{a}), Q(s_1, \mathrm{b}))$ into $\mathcal{Q}$
end for
$N \leftarrow$ size(uniq($\mathcal{Q}$))
if $N$>1 then
    More than 1 fixed points from given initial Q pairs.
else
    No more than 1 fixed points from given initial Q pairs.
end if
```

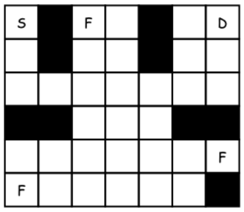
3.1.3 Ramdom MDP - GVI

The following pseudocode shows how we run generalized value iteration (GVI) in the randomly constructed MDP environtment to find the number of fixed points. Similar to 3.1.2, here we decide if > 1 fixed points by sampling initial Q values. Hence, it's more accurate if more various Q initial values are tried. (It explains this more here [4.1.4].)

Algorithm 3 Ramdom MDPs - GVI

```
Input: softmax operator $\otimes$ (bolz or mm), $N_{\text{MDP}}$, $N_{\text{trial}}$, $\alpha$ and $\beta$ or $\omega$
function CONSTRUCTMDP
    $\mathcal{P} : |\mathcal{S}| \times |\mathcal{A}| \times |\mathcal{S}|; \mathcal{R} : |\mathcal{S}| \times |\mathcal{A}|$
    $|\mathcal{S}|$ sample from $\{2, 3, 4, 5\}$ uniformly at random.
    $|\mathcal{A}|$ sample from $\{2, 3, 4\}$ uniformly at random.
    $\mathcal{P} \sim U[0, 0.01], \mathcal{R} \sim U[0, 0.01]$
    Each entry of $\mathcal{P}$ and $\mathcal{R}$ increase a value $\sim \mathcal{N}(1, \sqrt{0.1})$ with prob. 0.5.
    Each entry of $\mathcal{P}$ and $\mathcal{R}$ increase a value $\sim \mathcal{N}(100, \sqrt{1})$ with prob. 0.1.
    Normalize $\mathcal{P}$ to be a transition matrix.
    For each $s$ of $\mathcal{R}$, divide $\mathcal{R}(s, \cdot)$ by $\max(\mathcal{R}(s, \cdot))$ and $\times 0.5$.
end function
$N_{\text{nonSingle}} \leftarrow 0$
$\delta \leftarrow 10^{-14}$
for $i \leftarrow$ to $N_{\text{MDP}}$ do
    $\mathcal{Q} \leftarrow \{\}$                              ▷ $\mathcal{Q}$ stores each trial's convergence fixed point.
    for $j \leftarrow$ to $N_{\text{trial}}$ do
        All value $Q \sim U[0, 30]$
        repeat
            for $s \in \mathcal{S}$ do
                for $a \in \mathcal{A}$ do
                    $Q_{\text{copy}} \leftarrow Q(s, a)$
                    $Q(s, a) \leftarrow R(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s, a, s') \otimes Q(s', \cdot)$
                    $\text{diff}_{s,a} \leftarrow \max(\text{diff}_{s,a}, |Q_{\text{copy}} - Q(s, a)|)$
                end for
            end for
        until $\max(\text{diff}_{s,a}) < \delta$
        Push fixed point $Q$ into $\mathcal{Q}$
    end for
    $N_{\text{fixedPoints}} \leftarrow$ size(uniq($\mathcal{Q}$))
    if $N_{\text{fixedPoints}} > 1$ then
        $N_{\text{nonSingle}} \leftarrow N_{\text{nonSingle}} + 1$
    end if
end for
Average number of cases that > 1 fixed points of all MDPs' all trials is $\frac{N_{\text{nonSingle}}}{N_{\text{MDP}} \times N_{\text{trial}}}$.
```

3.2 Taxi Domain



environment (Dearden et al. [1998]): In this experience, s is the starting position, d is the destination, and f represents passenger, it will get +1 reward for delivering one passenger, +3 for two , +15 for three.

we preform SARSA with epsilon-greedy, SARSA with Boltzmann softmax and SARSA with Mellowmax softmax. In this environment, there are 33 posible positions and three passengers, therefore, it has 33*2*2*2 = 264 states and four possible actions.

In our algorithm, we set some prohibited state-action pair's Q values to negative infinity, like hit the wall. Moreover, durning the training period, we will not choose those action in order not to affect other Q values.

Because this environment need to have proper exploration, therefore, Boltzmann and Mellowmax can show their advantage. However, there are some detail that not be mention in this paper, so we have our own setting.

In epsilon-greedy method, training and testing both use epsilon-greedy to select action. And in Boltzmann softmax, training and test both use Blotzmann too. However, in Mellowmax softmax, we use Mellowmax softmax durning training process, and in testing process, we select the actions which have max Q values according to our pre-train Q values. We found that there are more approximate to paper's results.

3.3 Lunar Lander Domain

Experiment Settings

The paper's settings:

- network: a hidden layer comprised of 16 units with RELU activation functions + a second layer with 16 units and softmax activation functions
- use REINFORCE to train the network
- batch episode size: 10
- learning rate = 0.005
- optimizer: Adam
- do 10 experiments
  - Boltzmann softmax: $\beta = 1, 2, 3, 5, 10$
  - Mellowmax: $\omega = 3, 5, 7, 8, 11$
- each experiments do 400 runs, each runs train 40000 episodes

Our expiriment settings are basically the same as the paper's, the only different is we do each experiment 3 runs (seed=22, 321, 7654), each run trains 15000 episodes. Because the training needs a lot of time, so we simplified this part.

4 Empirical Evaluation

4.1 Simple MDP and Random MDPs

The author of this paper construct a handcrafted simple MDP, as shown in figure 1. It is composed of two states s_1 and s_2. State s_2 is a terminal state, and only s_1 is a non-terminal state. The edges are labeled with a transition probability (unsigned) and a reward number (signed). Discount factor $\gamma = 0.98$. State s_1 has two actions a and b. In the following simple MDP experiments, we mainly focus on the Q value of a and b.

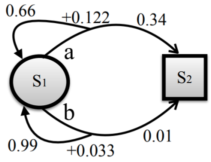
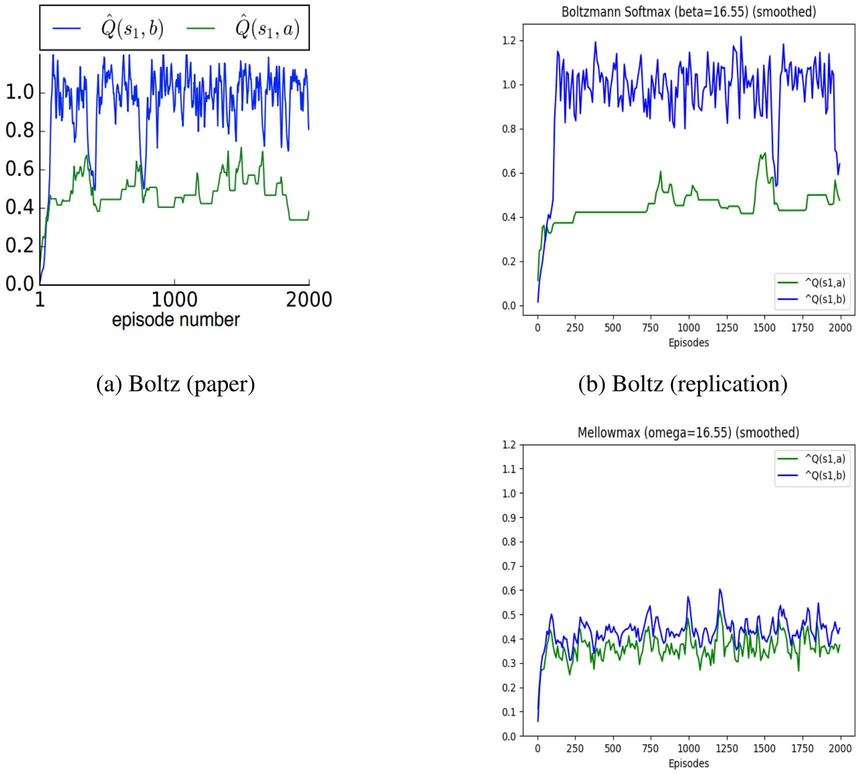


Figure 1: Simple MDP

4.1.1 Simple MDP - SARSA

Figure 2 shows the results of SARSA (algorithm) running 2000 episodes in this MDP. The blue and green lines depict the Q-values for both actions in each episode. Boltzmann curves have significantly larger oscillations. Its swing range is around 0.3 to 0.4, while the Q value of the Mellowmax curve is relatively stable; its swing range is around 0.2.



(a) Boltz (paper)

(b) Boltz (replication)

(c) Mellowmax (paper does not has this)

(d) Mellowmax (our)

Figure 2: SARSA with bolz/mm policy ($\beta = 16.55, \omega = 16.55$)

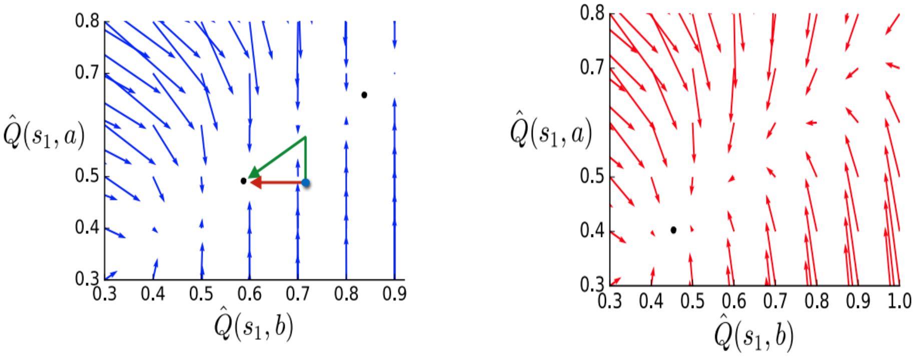
4.1.2 Simple MDP - GVI

Compared with value iteration, GVI doesn not restrict the way to compute $\otimes Q(s', \cdot)$. Max, Boltzmann Softmax, Mellowmax, and other operators can be used as an operator. (For more detail of GVI, please refer toLittman and Szepesvári [1996].)

Figure 3 (from paper) demonstrates the GVI result of Boltzmann Softmax and Mellowmax. An arrow is the updating direction and step size of a Q pair. Black points are the convergence fixed points. As we can see, in this case, there are two convergence points of Boltzmann Softmax's, while Mellowmax has only one convergence point.

Besides, the arrows on the figure tell us that the updating size is small when the Q pair locates at the position near the convergence point. This phenomenon is more obvious at the position between two convergence points especially. Thus, when there are more than one fixed points, the value may stay in the area between two convergence points with a tiny updating size, needing more iterations to reach a convergence point.

Boltzmann Softmax does not have the non-expansion property, so it is not under the convergence guarantee from the original GVI paper (Littman and Szepesvári [1996]). If there are > 1 convergence fixed points, the noise may drive it to swing between different points and lead to instability.

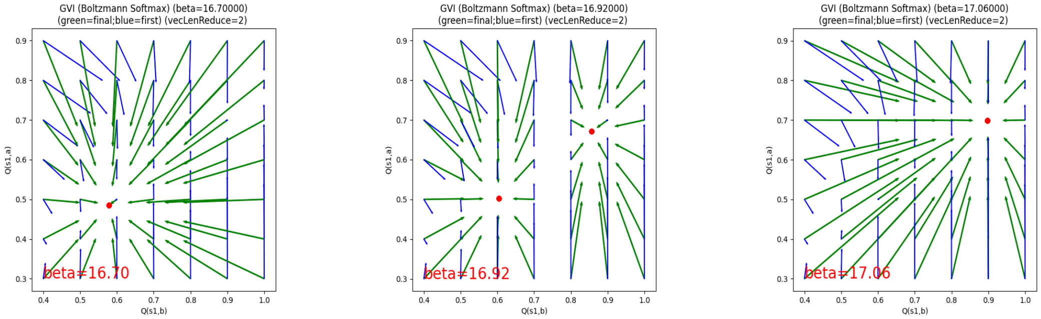


(a) Boltz (paper)

(b) Mellowmax (paper)

Figure 3: Paper's GVI with bolz/mm policy ($\beta = 16.55$, $\omega = 16.55$)

Figure 4 is our replication of GVI with Boltzmann Softmax policy under different hyper-parameter β. A blue arrow displays the first updating direction and relative size of one initial point. A green arrow displays the direction and relative updating size from starting point to convergence point. A red/balck point is a convergence fixed point. A range of β is tried, and we find under some β, there more than one convergence points as the figures shown.



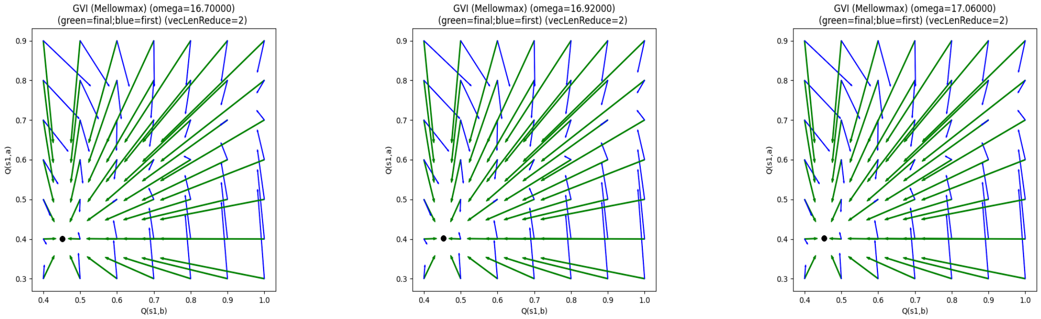
(a) $\beta = 16.7$

(b) $\beta = 16.92$

(c) $\beta = 17.06$

Figure 4: Our Boltzmann Softmax GVI replication

In conparison, Mellowmax does not show that it has more than one fixed points in our experiment. It's relative stable. We try to change ω from around 16.7 to 17, and the fixed point only move for a very small distance and stay near the point, as figure 5 displays.



(a) $\beta = 16.7$

(b) $\beta = 16.92$

(c) $\beta = 17.06$

Figure 5: Our Mellowmax GVI replication

The exact value of beta when it does not converge at only one point derived by us is not precisely equivalent to that shown on paper. But both of it show GVI with Boltzmann Softmax under a period of β near 16.7 converge at more than one points.

The following figure is from paper and it is obtained by trying different betas for GVI using the Boltzmann Softmax policy. Similar to the results in Figure 3 and 4, there are more than one convergence fixed point under some β.

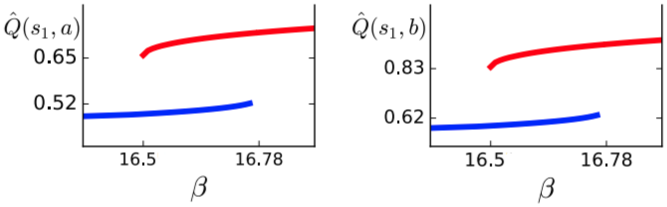
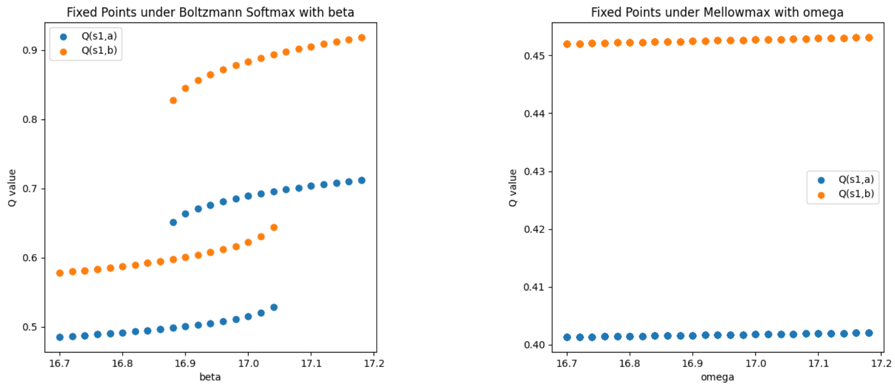


Figure 6: Number of different fixed points under different β (GVI with boltz)

Figure 7 is our result. Subfigure 7a shows fixed points of GVI using Boltzmann Softmax under different β. Although the precise value is not completely the same as figure 6, it demonstrates the same result as figure 6-there are more than one convergence fixed points under some β.

We make an additional subfigure 7b to show Mellowmax's result. Akin to what figure 5 shows, the fixed point of Mellowmax doesn't change a lot under different hyper-parameter ω. By comparison, it can be found that the convergence point of Boltzmann Softmax is more sensitive to hyperparameter tuning, while the convergence point of Mellomax is less affected by hyperparameter changes.



(a) Boltzmann Softmax

(b) Mellowmax

Figure 7: Our replication

4.1.3 Random MDPs

The experiments discussed so far were run in a handcrafted MDP environment. To see if these properties also exist under natural conditions, the paper conducts experiments on random MDPs to observe if randomly generated MDPs using these two policies for GVI also have these properties.

Figure 8 is the result from paper. It tries 200 MDPs. No termination means the GVI cannot converge within a predefined max number of iterations. Boltzman Softmax has these results: non-termination and >1 fixed points, while Mellowmax does not. In addition, Boltzmann Softmax requires more iterations on average to converge.

	MDPs, no terminate	MDPs, $>$ 1 fixed points	average iterations
boltz$_\beta$	8 of 200	3 of 200	231.65
mm$_\omega$	**0**	**0**	**201.32**

Figure 8: Random MDPs (paper). Max iteration is set 1000.

Figure 8 is our result. The way we construct random MDPs is a little different from the paper. See Algorithm 3 for details. We tried 200 MDPs; 100 trials were run for each MDP. The values of no-termination and average iteration in the figure are the results of averaging all MDPs' all trials. The values of > 1 fixed points are the results of averaging all MDPs' number. It can be seen that it presents the same characteristics as in Figure 8.

	avg # no terminate	avg # > 1 fixed points	avg interation
boltz$_\beta$	0.00675	0.03	1085.0973
mm$_\omega$	0	0	1048.794

Figure 9: Random MDPs (our). Max iteration is set 2000. There are 200 MDPs tried; each MDP has 100 trials.

4.1.4 How to determin wheter > 1 fixed points?

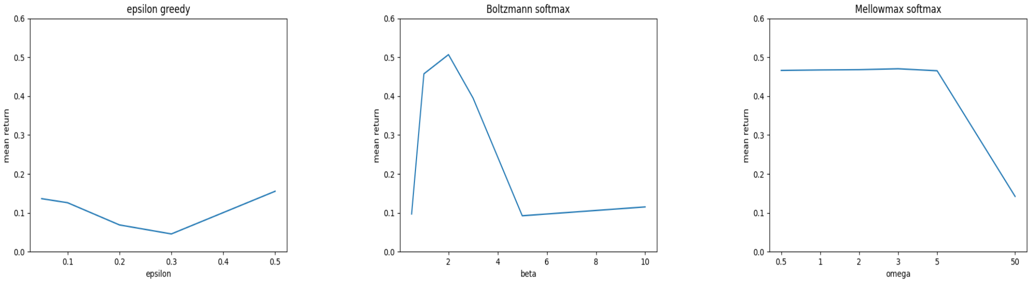
Note that the paper doesn't explain how they determine whether > 1 fixed points. The approach we use to decide if > 1 fixed points is to sample different initial Q values in each trial to see if there are different fixed points in all trials. That's why we run 100 trials for each MDP in figure 9. Although getting no > 1 fixed points by this method doesn't means it must actually have no > 1 fixed points, at least, we can derive it has a higher probability or less probability to have > 1 fixed points. For example, in our experiment result, Mellowmax never has > 1 fixed points among all trials, while Boltzmann has. We can infer that Boltzmann has at least a higher probability to have > 1 fixed points than Mellowmax.

4.2 Taxi Domain

Sarsa with epsilon-greedy, learning rate $\alpha = 0.1$, discount factor $\gamma = 1$, we test different ϵ values, $\epsilon = 0.05, 0.1, 0.2, 0.3, 0.5$, and the result are averaged over 6 independent runs, each consisting of 100000 time steps.

Sarsa with Boltzmann softmax, learning rate $\alpha = 0.1$, discount factor $\gamma = 1$, we test different β values, $\beta = 0.5, 1, 2, 3, 5, 10$, and the result are averaged over 6 independent runs, each consisting of 100000 time steps.

Sarsa with Mellowmax softmax, learning rate $\alpha = 0.1$, discount factor $\gamma = 1$, we test different ω values, $\omega = 0.5, 1, 2, 3, 5, 50$, and the result are averaged over 6 independent runs, each consisting of 10000 time steps. We know that when $\omega \to \infty$, Mellowmax will behave like max operator, and we set ω to 50 so that the result will similar to the paper and show that phenomenon.



(a) Boltz (paper)

(b) Boltz (paper)

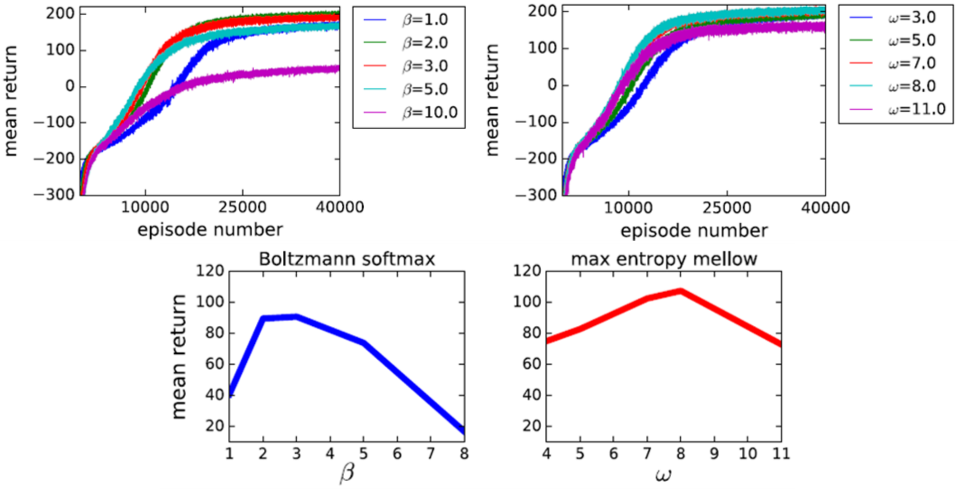
(c) Boltz (paper)

Figure 10: Taxi domain result

4.3 Lunar Lander Domain

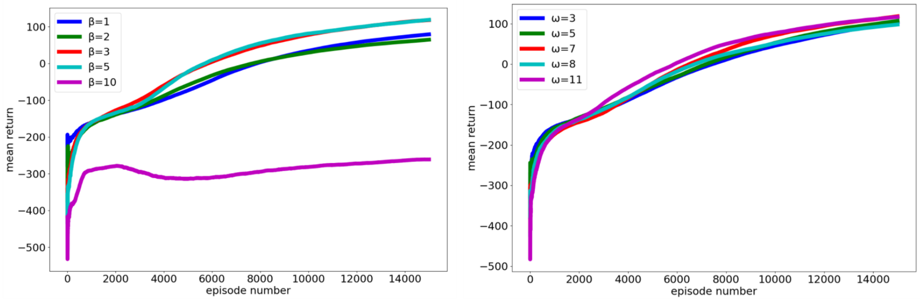
Experiment Results

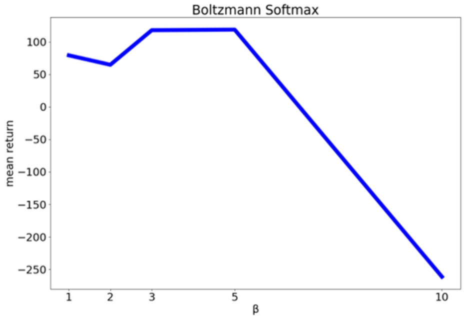
The paper's results:

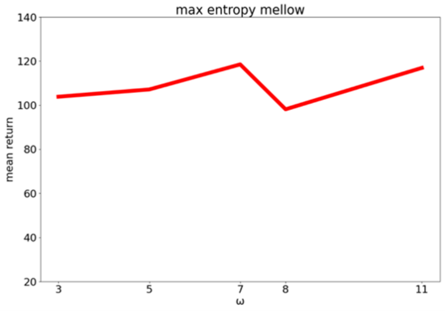


In the paper's results, the Boltzmann softsmax method performs well when $\beta = 2, 3$, and the Mellowmax method performs well when $\omega = 7, 8$.The Boltzmann softmax method, $\beta = 10$ performs the worst in the 10 experiments, it gets a much lower average mean return than others.The Mellowmax method, $\omega = 8$ performs the best in the 10 experiments.

Our results:







In our results, the Boltzmann softmax method performs well when $\beta = 3, 5$, and the Mellowmax method performs well when $\omega = 7, 11$. The Boltzmann softmax method, $\beta = 10$ performs the worst in the 10 experiments and gets a much lower average mean return than others. The Mellowmax method, $\omega = 7$ performs the best in our 10 experiments.

Comparison

Our experiment results are a little different from the paper's, and we think it is because that we use much less runs and less episodes in each run. The mean return of Boltzmann softmax $\beta = 10$ is about -300, and it's much lower than the paper's result. We find that it gets a lot of negative rewards at seed=22, its ewma reward is less than -1000 during almost all episodes, however, its ewma reward reaches 200 while seed=7654. It seems that Boltzmann softmax $\beta = 10$ is unstable, so the chosen seed affects a lots, and more runs are needed to get precise results.

5 Conclusion

The whole survey and replication through the paper could summarize the advantages of Mellowmax in the following points:

- The non-expansion property: the convergence of GVI is guaranteed
- more stable during the training

and the disadvantages of Mellowmax is that:

- Using Mellowmax operator may need more time to solve beta, but may get better updates during training. In LunarLander, it takes about 0.0014 sec per step with Boltzmann softmax and 0.0046 sec per step with Mellomax.

This paper proposed the Mellowmax operator as an alternative to the Boltzmann softmax operator. We also replicated that mellowmax has several desirable properties and that it works favorably in practice. Arguably, mellowmax could be used in place of Boltzmann throughout reinforcement-learning research.

References

Kavosh Asadi and Michael L Littman. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, pages 243–252. PMLR, 2017.

Richard Dearden, Nir Friedman, and Stuart Russell. Bayesian q-learning. *Aaai/iaai*, 1998:761–768, 1998.

Michael L Littman and Csaba Szepesvári. A generalized reinforcement-learning model: Convergence and applications. In *ICML*, volume 96, pages 310–318. Citeseer, 1996.